In the Specification:

Please amend paragraph 30 to read as follows:

[0030] By way of example, the time code and the read time data sets are time-locked during the time period associated with the second and third records of table 36, wherein the bottommost record is the first record and the records are numbered in an ascending fashion such that the topmost record is the sixth record, because the read time interval, Δtr 44 between these records is approximately equal to the time code interval Δtc 46 associated with these records. In contrast, the data sets are not time-locked during the time period associated with the fourth record 38 because the corresponding read time interval Δtr 44 and time code interval Δtc 46 are not approximately equal. Specifically, a time code 40 associated with the fourth record 38 is earlier than the time code 42 of the previous, third record 38 thereby causing the read time interval 44 and time code interval 46 associated with the fourth record to be mismatched or unequal. The read time interval 44 is approximately equal to the corresponding time code interval 46 for the fifth and sixth records such that the data sets are again ~~time locked~~ time-locked.

Please amend paragraphs 34-37 as follows:

[0034] Referring also to FIG. 4B which aligns with FIG. 4A at connection points A, B, C and D, if the SID values match, i.e., $SID_{test} = SID_1$, then the time code, tc_1, is associated with the same broadcast program as tc_{test} such that tc_1 may be used to validate tc_{test}. As a result, the method 50 continues at a step 68 where time interval data is calculated. More particularly, at the step 68, an interval denoted Δtc between the time codes, tc_{test} and tc_1, is calculated and an interval denoted Δtr between a read time denoted tr_{test} that corresponds with the time code tc_{test} and a read time denoted ~~tr_{test}~~ tr_1 that corresponds with tc_1.

[0035] Next, at a step 70, the interval Δtr is compared to the interval Δtc. If the interval Δtc is equal to the interval Δtr, within an allowable tolerance (TOL), i.e., ~~$\Delta tc = \Delta tr \pm TOL$~~ $\Delta tr - TOL < \Delta tc < \Delta tr + TOL$, then both Δtc_{test} and Δtc_1 are valid and are marked valid at a step 72. For example, the time codes Δtc_{test} and Δtc_l and Δtc_l may be marked valid by setting a validity flag associated with each value. Alternatively, any data association method may be used to indicate that the values Δtc_{test} and Δtc_l and Δtc_l are valid.

[0036] If instead the interval Δtc is not equal to the interval Δtr, within an allowable tolerance (TOL), i.e., ~~$\Delta tc \neq \Delta tr \pm TOL$~~ $\Delta tc > \Delta tr + TOL$ or $\Delta tc < \Delta tr - TOL$, then one or both of Δtc_{test} and Δtc_l are invalid and the method branches back to the step 60 and the steps subsequent thereto, as described above. Likewise, after the values Δtc_{test} and Δtc_l have been marked valid at the step 72, the method branches back to the step 60.

[0037] As described above, if, at the step 62, COUNTER2 is greater than COUNTER1, then all of the time codes in the stack have been compared to ~~Δtc_1~~ Δtc_{test} and the method 50 continues at the step 74 where the processor 32 determines whether tc_{test} has been validated by the portion of the method 50 including the steps 60-72. If tc_{test} has not been validated, thereby indicating that tc_{test} is erroneous, then the method continues at the step 76 where tc_{test} is corrected. Specifically, at the step 76, SID_{test} is compared to an SID, denoted $SID_{previous}$, that corresponds to the most recently validated time code denoted

$tc_{previous}$. The value of $tc_{previous}$ is equal to the value of the time code that was most recently removed from the stack for processing by the method 50 and that was validated during the most recent of the previous iterations of the method 50. If the SID values are equal, i.e., $SID_{test}= tc_{previous}$ thereby indicating that the time codes, tc_{test} and $tc_{previous}$, were extracted from the same broadcast program, then a read time interval $\Delta tr_{previous}$ between a read time denoted $tr_{previous}$ that corresponds to the time code, $tc_{previous}$, and the read time, tr_{test}, that corresponds to the time code tc_{test} is calculated at a step 78. At the step 78, the read time interval $\Delta tr_{previous}$ is also added to the value of $tc_{previous}$ to obtain a corrected value for the time code tc_{test} which is subsequently stored in a list of validated time codes, at a step 80, for later retrieval and usage in calculating corrected time codes as necessary. After storing the value of tc_{test}, the method branches back to the step 54 where the next time code value is extracted from the bottommost position in the stack and the method continues at the steps subsequent thereto as described above.